BESCHWERDEKAMMERN
DES EUROPÄISCHEN
PATENTAMTS

BOARDS OF APPEAL OF
THE EUROPEAN PATENT
OFFICE

CHAMBRES DE RECOURS
DE L'OFFICE EUROPÉEN
DES BREVETS

Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 21 November 2022

Case Number:	T 3176/19 - 3.5.07
Application Number:	15174381.2
Publication Number:	3113038
IPC:	G06F17/30, G06F9/54
Language of the proceedings:	EN

Title of invention:
A data handling method

Applicant:
Broadridge Financial Solutions Limited

Headword:
Data handling system/Broadridge Financial Solutions

Relevant legal provisions:
EPC Art. 56, 84, 123(2)

Keyword:
Claims - clarity (yes)
Amendments - added subject-matter (no)
Inventive step - (yes)

Decisions cited:
T 1351/04

This datasheet is not part of the Decisior
It can be changed at any time and without notic

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 3176/19 - 3.5.07

DECISION
of Technical Board of Appeal 3.5.07
of 21 November 2022

Appellant:
(Applicant)

Broadridge Financial Solutions Limited
Isis Building
Thames Quay
193 Marsh Wall
London E14 9SG (GB)

Representative:

Ahmad, Sheikh Shakeel
Keltie LLP
No.1 London Bridge
London SE1 9BA (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 11 July 2019 refusing European patent application No. 15174381.2 pursuant to Article 97(2) EPC

Composition of the Board:

Chair J. Geschwind
Members: P. San-Bento Furtado
R. de Man

Summary of Facts and Submissions

I. The appeal lies from the decision of the examining division to refuse European patent application No. 15174381.2.

The following documents were cited in the decision under appeal:

D1: US 5819282A, published on 6 October 1998; and

D2: J. Winchester et al: "XML Serialization of Java Objects", Java Developer's Journal, June 2003 retrieved from http://java.sys-con.com/node/37550.

The decision also refers briefly to a document D3.

II. The examining division decided that the subject-matter of claim 1 of the sole request was not clearly defined and lacked inventive step over prior-art document D1 in combination with the common general knowledge of the skilled person. In addition, certain distinguishing features were considered to be disclosed in documents D2 and D3. The subject-matter of the dependent claims was not inventive because the additional features specified in those claims were normal design options or obvious implementation details.

III. With the statement of grounds of appeal, the appellant maintained, as main request, the set of claims considered in the decision under appeal and filed claims according to four auxiliary requests. The appellant contested the mapping of features in the appealed decision's assessment of inventive step over document D1.

IV. In a communication accompanying a summons to oral proceedings, the board expressed its preliminary opinion that claim 1 of each of the requests lacked clarity, that the inventive-step reasoning of the decision under appeal was not convincing and that document D1 was not a suitable starting point. The board compared the claimed subject-matter with the prior-art data handling system acknowledged in the background section of the application, which was considered to be a more suitable starting point.

V. With a letter of reply, the appellant filed amended claims of a new main request B.

VI. Oral proceedings were held as scheduled. During the oral proceedings the appellant submitted an amended set of claims of a new main request C replacing all the other requests. The appellant further reverted to the description pages as originally filed.

VII. The appellant's final requests were that the decision under appeal be set aside and that a patent be granted on the basis of the claims of main request C and the description and drawings as originally filed.

VIII. Claim 1 of the sole request, main request C, reads as follows:

"A data processing system comprising a target data structure (100) and a data processing engine (200) for processing the target data structure (100), the data processing engine (200) comprising a plurality of different processing components (38, 40), each processing component (38, 40) being arranged to perform a different processing operation on the target data structure,

the target data structure (100) comprising a plurality of data fields (102), each of the data fields (102) having a corresponding index (104), and each of the data fields (102) comprising a field name (106) and a data value (108), wherein the target data structure (100) is a Java object in the form of a Collection data structure,

the data processing engine (200) further comprising:
an internal topic bus (46) connected to the plurality of processing components (38, 40);
a marshalling component (34) arranged to generate a serial representation of the target data structure (100) obtained from an external message bus (32) to marshal the target data structure (100) into a marshalled format from an unmarshalled format, the marshalling component (34) being arranged to place the marshalled format of the target data structure (100) onto the internal topic bus (46) such that the target data structure (100) can be conveyed to each of the plurality of processing components (38, 40) and processed by the same; and,
an unmarshalling component (36) arranged to perform a converse transformation back from the serial representation to unmarshal the processed target data structure (100) from the marshalled format on the internal topic bus (46) to the unmarshalled format, and the unmarshalling component (36) being arranged to output the processed target data structure (100) onto the external message bus (32),

each of the processing components (38, 40) of the data processing engine (200) comprising:

a receiver arranged to receive configuration data comprising configuration tasks which each define a predetermined set of prioritised field names (106), wherein the predetermined set of prioritised field names is a predetermined subset of the total possible field names (106) of the target data structure (100) including at least one prioritised field name and an index identifier corresponding to each respective prioritised field name;

a key generator arranged to establish, for each prioritised field name in the predetermined set of field names (106), an access key for directly accessing a specific location of the target data structure using the index identifier corresponding to the respective prioritised field name, wherein each of the access keys is a Java object or subroutine, and wherein the specific location is the index (104) of the target data structure (100) corresponding to the index identifier of the respective access key, the key generator being arranged to store the access keys in an accessor database of the respective processing component; and,

a component-specific processor (48, 50) arranged to: obtain data from the internal topic bus (46) and read the obtained data into the target data structure (100) using the access keys from the accessor database to determine what index position to place the obtained data in; read the data values (108) of respective data fields (102) directly using the access keys when the data fields (102) to be read correspond to one of the prioritised field names in the predetermined set, the component-specific processor (48, 50) being arranged to iterate through the target data structure to find additional data fields not corresponding to one of the prioritised field names if the additional data fields are required to be read; and, the component-specific processor (48, 50) being arranged to generate an output based on the data values (108) of the data fields (102) to be read, wherein the output is generated by carrying out calculations on the data values of the data fields (102) to be read to generate the output,

wherein the prioritised field names in the predetermined set is a subset including the field names of the data fields that are determined to be the most commonly accessed data fields in the processing carried out across all of the components of the data processing engine."

IX. Claims 2 to 7 are directly or indirectly dependent on claim 1.

Reasons for the Decision

Documents D3

1. According to the contested decision, document D3 had been cited in the communication of 13 June 2017, but document D3 cited in the decision under appeal (US 4 841 433 of 20 June 1989) is not the same as document D3 cited in that communication (US 5 542 089 of 30 July 1996). Since in the contested decision the examining division did not cite any specific passage of the document, it is unclear to which document D3 the decision under appeal was referring. In this decision, both documents D3 are considered.

Application

2. The present patent application concerns a technique to support efficient data processing in a processing system by means of a new data structure (see page 1,

lines 5 to 8; page 4, line 23, to page 5, line 16 of the description as filed).

2.1 In its background section, the application describes a prior-art processing system, which is illustrated in Figure 3:

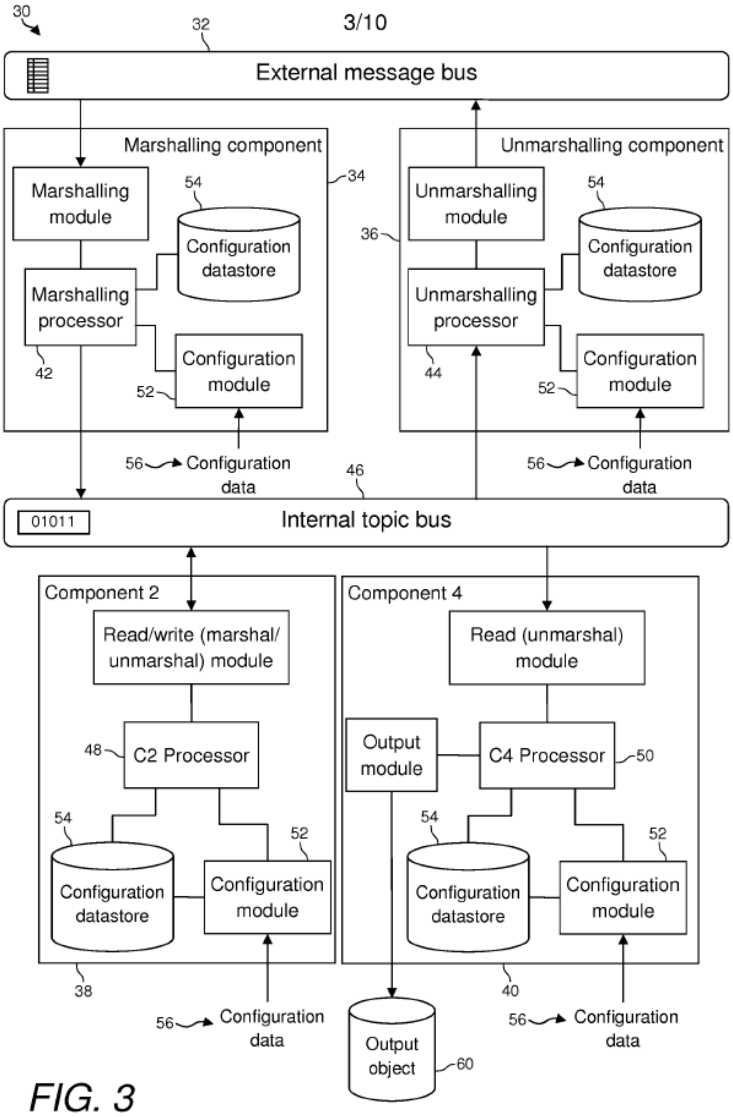

FIG. 3

The processing system comprises four components 34, 36, 38 and 40, all connected to an "internal topic bus" 46. The marshalling and unmarshalling components 34 and 36 are connected to an "external message bus" 32, which is "a communication channel on which real-time information is provided about an entity". The marshalling component 34 generates a serial representation of the data in a common data format (typically an XML format) and outputs it to the internal topic bus. The unmarshalling component 36 performs the converse operation. According to the description, these two components can be implemented by means of the JAXB Java API (page 2, line 18, to page 3, line 8; Figure 3).

The other two components 38 and 40 perform different operations on the data. Each of these processing components is configured to look for a specific type of data structure (Collection) on the internal topic bus and process the data found (page 3, line 10, to page 4, line 4).

2.2 The prior art processing system uses two known Java Collection data structures, the ArrayList, with a sequential index, and the HashMap, which generates an index by applying a hash function to a key value (page 1, line 18, to page 2, line 16; Figures 1 and 2). An ArrayList implementation of a Collection requires sequentially iterating through the entries of the Collection even if only a few items are being read or written to (page 1, lines 30 to 33; page 4, lines 9 and 10). A HashMap implementation is inefficient when a plurality of data fields is accessed (page 2, lines 7 to 13, page 4, lines 10 to 13).

2.3 The processing system of the invention, illustrated in Figure 5, adds an "accessor data store" 202 ("accessor database" in the claims) to each of the processing components 38 and 40.

In the system according to the invention, a "weighted data structure" is used, which is a Java Collection, with typically 150 to 200 data fields, each data field having a corresponding index and comprising a field name (i.e. a key) and a data value, as illustrated in Figure 4. A subset of data fields with frequently accessed data values are "prioritised data fields" and are assigned predetermined indices in the data structure (in the example of Figure 4, the shaded fields using the first six indices). If these data fields are absent in a particular instance of a data structure read from the internal topic bus 46, their values are set to "null" in the weighted data structure. For a field name of one of the prioritised data fields, processes use the corresponding index to directly access the data value, thereby trading storage space for data access time. For field names not corresponding to one of the prioritised data fields, the process iterates through each data field that is not in the set of prioritised data fields until it finds the desired data field (page 10).

2.4 Each modified processing component 38 and 40 receives configuration data comprising configuration tasks which each define a pre-determined set of field names relating to prioritised data fields. The processors 48 and 50 of the components interpret each configuration task and create Java objects or subroutines that provide direct access to a specific location (index) of the data structure without requiring any search of the data (page 11, lines 1 to 20; page 12, lines 5 to 8). The created configuration tasks are stored in the accessor datastore (page 11, line 20; page 12, lines 8 to 10).

Admission into the appeal proceedings of the sole request

3. In its communication accompanying the summons to oral proceedings the board raised new clarity objections and did not follow the inventive-step assessment of the

decision under appeal. In reply to this communication, the appellant filed a new set of claims, as main request B, dealing with those objections. Since main request B was filed in response to objections which had been raised for the first time in the board's communication, its admission into the appeal proceedings was justified by an exceptional circumstance, as required by Article 13(2) RPBA 2020.

4. The current set of claims, main request C, is based on main request B. It was filed during the oral proceedings and the board found that it *prima facie* overcame all the remaining objections, including new objections under Article 84 EPC raised by the board during the oral proceedings. Therefore, its admission into the appeal proceedings was justified by an exceptional circumstance, as required by Article 13(2) RPBA 2020.

Clarity and support - Article 84 EPC

5. Claim 1 considered in the decision under appeal has been redrafted to overcome objections of lack of both clarity and conciseness raised by the board. Claim features have been reordered and repetitions have been eliminated. In addition, the claim has been amended to delete the text "interpret each of the received configuration tasks ... without requiring any search of the target data structure" and to specify the following features:

(a) "each of the data fields having a corresponding index, and each of the data fields comprising a field name and a data value, wherein the target data structure is a Java object in the form of a Collection data structure,"

(b) "the predetermined set of prioritised field names is a predetermined subset of the total possible field names of the target data structure"

(c) "each of the access keys is a Java object or subroutine, and wherein the specific location is the index of the target data structure corresponding to the index identifier of the respective access key, the key generator being arranged to store the access keys in an accessor database of the respective processing component;"

5.1 The objections under Article 84 EPC raised against claim 1 in the contested decision and in the board's communication either no longer apply or have been overcome by these amendments.

5.2 No objections under Article 84 EPC have been raised against the dependent claims.

5.3 Therefore, the board is satisfied that the claims meet the requirements of Article 84 EPC.

Added subject-matter - Article 123(2) EPC

6. The subject-matter of claim 1 combines features of original claim 1, and features disclosed in Figures 4 to 6 and in the originally filed description on page 2, line 18 to page 3, line 9, page 5, lines 4 to 16, page 8, lines 1 to 19, page 10, lines 1 to 6 and 29 to 32, page 11, page 12, lines 5 to 10 and 17 to 23 and page 13, lines 7 to 14.

6.1 Feature (a) is disclosed on page 10, lines 1 to 6 of the description and Figure 4 as originally filed.

6.2 The basis for feature (b) can be found on page 11, lines 12 to 15 and page 12, lines 6 to 8 of the description as originally filed.

6.3 Feature (c) can be directly and unambiguously derived from the description on page 11, lines 17 to 20, in combination with page 8, lines 8 to 15 and page 12, lines 5 to 15.

7. Dependent claims 2 to 7 are based on dependent claims 2 to 5, 8 and 9 as originally filed.

8. Therefore, the board is satisfied that the claims meet the requirements of Article 123(2) EPC.

Inventive step - document D1

9. In the decision under appeal, the examining division assessed inventive step starting from document D1.

9.1 Document D1 discloses techniques for converting data from a database system into a structure for efficient processing of the data using object-oriented data manipulation (column 2, lines 4 to 26). It describes data structures and processes that provide an interface between relational databases and databases developed with object-oriented programming techniques (column 3, line 63, to column 4, line 2). A database relation such as EMPLOYEES, with attributes ID, NAME and LOCATION, is converted into a class with members ID, NAME and LOCATION stored in a CSV data structure named "expanded spread-sheet". The member names are used for referencing and locating data and processing information to the particular members of the class. The CSV file includes rows with data (D-rows) and rows with "processing information", for example the name of the class (C-row), attribute (i.e. member) names (M-row), accessing methods (A-row), type information (T-row) and queries (S-row). The data type of a member can be the name of a class storing data related to the thus typed member (column 4, line 3, to column 5, line 43; column 6, lines 8 to 20; Figure 5 and 6).

9.2 In its decision, the examining division considered that the feature "the target data structure comprising a plurality of data fields with corresponding indices, and a predetermined set of field names having a predetermined prioritised subset of data field names including at least one prioritised field name and a corresponding index identifier" was disclosed in document D1, column 4, lines 3 to 10, and Figure 2, and that the predetermined set of field names corresponded to the set of "sub-sets of data such as EMPLOYEES, SUPPLIERS and CUSTOMERS".

9.3 The appellant contested this mapping and the result of the novelty analysis of the decision under appeal. The examining division had not correctly interpreted claim 1, in that it had wrongly treated the receiver, key generator and component-specific processor as components of the data processing engine, even though they were part of each component. The mapping of the data structure was also incorrect.

9.4 The board agrees with the appellant that the classes of document D1 do not map to the predetermined data fields of claim 1 and that document D1 does not disclose prioritising of classes (nor of data fields within a class). In addition, document D1 does not disclose almost any of the structural aspects of the claimed processing system. In view of this, the inventive-step reasoning of the decision under appeal is not convincing and document D1 is not a suitable starting point for assessing inventive step.

Inventive step - acknowledged prior art

10. The prior-art data handling system acknowledged in the background section of the application, on page 2, line 18 to page 4, line 6 and Figures 1 to 3 (see also point 2.1 above) is a suitable starting point for assessing inventive step.

10.1 This system comprises a data processing engine, processing components, an internal topic bus and an external bus, as described in point 2.1 above. It supports processing of data structures, including marshalling, unmarshalling and processing the Collection data structures HashMap and ArrayList. The processing components are configured with instructions on how to iterate through an ArrayList and a HashMap (page 3, lines 18 to 25).

However, the system does not use a modified data structure in which a predetermined prioritised subset of data field names is established for the most commonly accessed data fields in the processing.

10.2 The system of claim 1 therefore differs from the acknowledged prior-art data handling system at least in the claim features specifying prioritised field names and respective access keys, and how they are generated and used. In particular, the prior-art system does not include a key generator nor configuration data comprising configuration tasks, each defining a predetermined set of prioritised field names.

10.3 By providing direct access to the most commonly accessed data fields, the claimed system improves the average access time required to access data fields when compared to the prior-art system. Therefore, the board agrees with the appellant that the new data structure improves the efficiency of the data processing system. This effect is also disclosed in the description (see e.g. page 1, lines 5 and 6; page 4, lines 8 to 25; page 14, line 13, to page 15, line 13). The board is further satisfied that the efficiency improvement, achieved by considering data access patterns, making a trade-off between storage space and time (see point 2.3 above), and using a data structure to determine the way the computer accesses information (see also T 1351/04, reasons 7.2 to 7.4) efficiently according to those access patterns is based on technical considerations relating to the functioning of a computer and is therefore a technical effect for the purpose of assessing inventive step.

In addition, by using the configuration data defining a predetermined set of prioritised field names, the solution of the invention is particularly adapted to the data handling system of Figure 3 in which data is processed by components connected to a common bus. The distinguishing features thus solve the technical problem of improving efficiency of data access in the particular context of the architecture of the prior-art data processing system.

11. The claimed solution is neither disclosed nor rendered obvious by the available prior art.

11.1 Document D2 is unrelated to this problem, and the solutions disclosed in the two documents D3 mentioned in point 1. above are very different from the present invention. These documents are therefore not prejudicial to inventive step of claim 1.

11.2 The decision under appeal argued that certain distinguishing features constituted "the normal means of accessing indexed or cached data".

However, in the board's opinion the prioritisation of data fields is not comparable to the function of a cache.

Even though indexing is well known, the idea of establishing prioritised field names and the particular use of access keys for the prioritised field names in the present invention is very different from the commonly known indexing solutions in a general purpose computer.

11.3 Document D1 discloses accessing methods used to support faster access to data in order to solve the same problem of improving data processing efficiency.

As the appellant argued, the data rows of the expanded spread-sheet of document D1 correspond to the data fields of claim 1. An A-row defines accessing methods for "direct access", which define how "data objects are located by means of access keys". A primary key (e.g. ID in Figure 5) is used for uniquely accessing the instances of data objects in the class. Document D1 discloses several accessing methods, including linked lists, hashing tables, and B-trees of primary or secondary keys. These provide efficient access to objects based on their keys (column 6, line 49 to column 7, line 56; Figure 8) in a conventional way, but not to data fields. Furthermore, they do not generate access keys as defined in claim 1.

Document D1 also discloses a query row (S-row) used to specify a pre-defined query, for example in SQL, on the objects of a class (column 6, lines 8 to 20) and accessing methods consisting of an array or list corresponding to a query. When such an accessing method is used, an array of memory addresses referencing all objects which satisfy the condition of the query is created and maintained up to date as the database is modified. The list of memory addresses is created to rapidly locate the qualified data objects (column 6, line 49 to column 7, line 56; Figure 8). Therefore, these data objects are "prioritised" within the class.

However, document D1 does not disclose using such an array to more efficiently access an individual object directly by its respective key (as specified in the distinguishing features). In the system of D1, the array is used to efficiently access all the objects satisfying the query by using the query name, not to access individual objects (column 6, lines 18 to 20; column 7, lines 31 to 47).

11.4 Taking this into account, the board is not convinced that the skilled person would, without inventive skill, have arrived at all the distinguishing features of claim 1.

11.5 The same applies to the dependent claims, which specify further features of the claimed data processing system.

11.6 The board concludes that the claims meet the requirements of Article 56 EPC.

Concluding remarks

12. The question of whether the description has to be adapted is moot since the description does not disclose any embodiments falling outside the scope of the invention as defined by the claims.

13. Since there are no remaining objections, the application is allowable.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the examining division with the order to grant a patent on the basis of the following documents:
   - claims 1 to 7 filed as main request C during the oral proceedings before the board;
   - description pages 1 to 15 as originally filed;
   - drawing sheets 1/10 to 10/10 as originally filed.

The Registrar: The Chair:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

S. Lichtenvort J. Geschwind

Decision electronically authenticated